

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

#3

J1046 U.S. PTO  
09/931937  
08/20/01

In re U.S. Patent Application of )  
KAMINAGA et al. )  
Application Number: To be assigned )  
Filed: Concurrently herewith )  
For: FAULT DETECTION METHOD )

Honorable Assistant Commissioner  
for Patents  
Washington, D.C. 20231

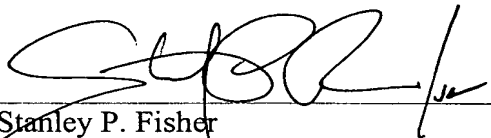
**REQUEST FOR PRIORITY  
UNDER 35 U.S.C. § 119  
AND THE INTERNATIONAL CONVENTION**

Sir:

In the matter of the above-captioned application for a United States patent, notice is hereby given that the Applicant claims the priority date of March 2, 2001, the filing date of the corresponding Japanese patent application 2001-058087.

The certified copy of corresponding Japanese patent application 2001-058087 is submitted herewith. Acknowledgment of receipt of the certified copy is respectfully requested in due course.

Respectfully submitted,



Stanley P. Fisher  
Registration Number 24,344

**REED SMITH HAZEL & THOMAS LLP**  
3110 Fairview Park Drive  
Suite 1400  
Falls Church, Virginia 22042  
(703) 641-4200  
August 20, 2001

**JUAN CARLOS A. MARQUEZ**  
Registration No. 34,072

日 本 国 特 許 庁  
JAPAN PATENT OFFICE

J1046 U.S. PTO  
09/931937  
08/20/01

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日

Date of Application:

2001年 3月 2日

出 願 番 号

Application Number:

特願2001-058087

出 願 人

Applicant(s):

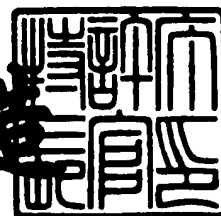
株式会社日立製作所

CERTIFIED COPY OF  
PRIORITY DOCUMENT

2001年 7月27日

特 許 庁 長 官  
Commissioner,  
Japan Patent Office

及 川 耕 造



出証番号 出証特2001-3066031

【書類名】 特許願

【整理番号】 NT00P1114

【提出日】 平成13年 3月 2日

【あて先】 特許庁長官 殿

【国際特許分類】 G06K 19/00

【発明者】

    【住所又は居所】 東京都国分寺市東恋ヶ窪一丁目 2 8 0 番地 株式会社日立製作所 中央研究所内

    【氏名】 神永 正博

【発明者】

    【住所又は居所】 東京都国分寺市東恋ヶ窪一丁目 2 8 0 番地 株式会社日立製作所 中央研究所内

    【氏名】 遠藤 隆

【発明者】

    【住所又は居所】 東京都国分寺市東恋ヶ窪一丁目 2 8 0 番地 株式会社日立製作所 中央研究所内

    【氏名】 渡邊 高志

【発明者】

    【住所又は居所】 東京都国分寺市東恋ヶ窪一丁目 2 8 0 番地 株式会社日立製作所 中央研究所内

    【氏名】 大木 優

【特許出願人】

    【識別番号】 000005108

    【氏名又は名称】 株式会社日立製作所

【代理人】

    【識別番号】 100068504

    【弁理士】

    【氏名又は名称】 小川 勝男

    【電話番号】 03-3661-0071

【選任した代理人】

【識別番号】 100086656

【弁理士】

【氏名又は名称】 田中 恭助

【電話番号】 03-3661-0071

【選任した代理人】

【識別番号】 100094352

【弁理士】

【氏名又は名称】 佐々木 孝

【電話番号】 03-3661-0071

【手数料の表示】

【予納台帳番号】 081423

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 暗号処理方法

【特許請求の範囲】

【請求項 1】

情報処理装置を利用して対称鍵暗号化処理を行なう方法であって、

- (1) 入力される平文Mに秘密鍵Kを適用する暗号化処理  $Z = E(M, K)$  を行なってその結果Zをメモリに格納し、
- (2) 前記メモリ上の結果Zに対して復号化処理  $W = D(Z, K)$  を行なってその結果Wをメモリ上に格納し、
- (3) 前記の処理結果Wと平文Mとが一致している場合には、処理結果Zを出力し、
- (4) 前記の処理結果Wと平文Mとが不一致の場合には、処理結果の出力を抑止することを特徴とする暗号処理方法。

【請求項 2】

前記暗号化処理及び復号化処理を D E S (Data Encryption Standard) に従って実行することを特徴とする請求項 1 記載の暗号処理方法。

【請求項 3】

前記処理結果の出力を抑止する方法として、前記情報処理装置をリセットすることを特徴とする請求項 1 記載の暗号処理方法。

【請求項 4】

前記情報処理装置及び前記メモリは、I C カード上に搭載されるそれぞれ演算装置及び記憶装置であることを特徴とする請求項 1 記載の暗号処理方法。

【請求項 5】

情報処理装置を利用して対称鍵復号化処理を行なう方法であって、

- (1) 入力される暗号文Cに秘密鍵Kを適用する復号化処理  $Z = D(C, K)$  を行なってその結果Zをメモリに格納し、
- (2) 前記メモリ上の結果Zに対して暗号化処理  $W = E(Z, K)$  を行なってその結果Wをメモリ上に格納し、
- (3) 前記の処理結果Wと暗号文Cとが一致している場合には、処理結果Zを出

力し、

(4) 前記の処理結果Wと暗号文Cとが不一致の場合には、処理結果の出力を抑止することを特徴とする暗号処理方法。

【請求項 6】

前記暗号化処理及び復号化処理を D E S (Data Encryption Standard) に従って実行することを特徴とする請求項 5 記載の暗号処理方法。

【請求項 7】

前記処理結果の出力を抑止する方法として、前記情報処理装置をリセットすることを特徴とする請求項 5 記載の暗号処理方法。

【請求項 8】

前記情報処理装置及び前記メモリは、I C カード上に搭載されるそれぞれ演算装置及び記憶装置であることを特徴とする請求項 5 記載の暗号処理方法。

【請求項 9】

情報処理装置を利用して非対称鍵復号化処理を行なう方法であって、

(1) 入力される暗号文Cに秘密鍵X、公開鍵情報Jを適用する復号化処理  $Z = D(C, X, J)$  を行なってその結果Zをメモリに格納し、

(2) 前記メモリ上の結果Zに対して暗号化処理  $W = E(Z, J)$  を行なってその結果Wをメモリ上に格納し、

(3) 前記の処理結果Wと暗号文Cとが一致している場合には、処理結果Zを出力し、

(4) 前記の処理結果Wと暗号文Cとが不一致の場合には、処理結果の出力を抑止することを特徴とする暗号処理方法。

【請求項 10】

前記暗号化処理及び復号化処理を R S A 暗号化方式に従って実行することを特徴とする請求項 9 記載の暗号処理方法。

【請求項 11】

前記処理結果の出力を抑止する方法として、前記情報処理装置をリセットすることを特徴とする請求項 9 記載の暗号処理方法。

【請求項 12】

前記情報処理装置及び前記メモリは、ＩＣカード上に搭載されるそれぞれ演算装置及び記憶装置であることを特徴とする請求項９記載の暗号処理方法。

【発明の詳細な説明】

【 0 0 0 1 】

【発明の属する技術分野】

本発明は、高いセキュリティを持つＩＣカードなどの耐タンパー暗号処理方法に関するものである。

【 0 0 0 2 】

【従来の技術】

ＩＣカードは、勝手に書き換えることが許されないような個人情報の保持や、秘密情報である暗号鍵を用いたデータの暗号化や暗号文の復号化を行う装置である。ＩＣカード自体は電源を持っておらず、ＩＣカード用のリーダライタに差し込まれると、電源の供給を受け、動作可能となる。動作可能になると、リーダライタから送信されるコマンドを受信し、そのコマンドに従ってデータの転送等の処理を行う。ＩＣカードの一般的な解説は、オーム社出版電子情報通信学会編水沢順一著「ＩＣカード」などにある。

【 0 0 0 3 】

ＩＣカードの構成は、図１に示すように、カード１０１の上に、ＩＣカード用チップ１０２を搭載したものである。図に示すように、一般にＩＣカードは、ＩＳ０７８１６の規格に定められた位置に供給電圧端子Ｖcc、グランド端子ＧND、リセット端子ＲST、入出力端子Ｉ／Ｏ及びクロック端子CLKを持ち、これらの端子を通してリーダライタから電源の供給やリーダライタとのデータの通信を行う(W.Rankl and E. ffinfing : SMARTCARD HANDBOOK、John Wiley & Sons、1997、pp.41参照)。

【 0 0 0 4 】

ＩＣカード用チップの構成は、基本的には通常のマイクロコンピュータと同じ構成である。その構成は、図２に示すように、中央処理装置(CPU)201、記憶装置204、入出力(I/O)ポート207、コ・プロセッサ202からなる(コ・プロセッサはない場合もある)。CPU201は、論理演算や算術演算などを行う装置であり、記憶装置204は、プログラムやデータを格納する装置である。入出力ポー

トは、リーダライタと通信を行う装置である。コ・プロセッサは、暗号処理そのもの、または、暗号処理に必要な演算を高速に行う装置であり、例えばRSA暗号の剰余演算を行うための特別な演算装置や、DES暗号のラウンド処理を行う暗号装置などがある。ICカード用プロセッサの中には、コ・プロセッサを持たないものも多くある。データバス203は、各装置を接続するバスである。

## 【0005】

記憶装置204は、ROM(Read Only Memory)やRAM(Random Access Memory)、EEPROM(Electrical Erasable Programmable Read Only Memory)などからなる。ROMは、変更できないメモリであり、主にプログラムを格納するメモリである。RAMは自由に書き換えができるメモリであるが、電源の供給が中断されると、記憶している内容は消滅する。ICカードがリーダライタから抜かれると電源の供給が中断されるため、RAMの内容は、保持されなくなる。EEPROMは、電源の供給が中断されてもその内容を保持することができるメモリである。書き換える必要があり、ICカードがリーダライタから抜かれても、保持するデータを格納するために使われる。例えば、プリペイドカードでのプリペイドの度数などは、使用するたびに書き換えられ、かつリーダライタから抜かれてもデータを保持する必要があるため、EEPROMで保持される。

## 【0006】

ICカードは、プログラムや重要な情報がICカード用チップの中に密閉されているため、重要な情報を格納したり、カードの中で暗号処理を行うために用いられる。従来、ICカードでの暗号を解読する難しさは、暗号アルゴリズムの解読の困難さと同じと考えられていた。

## 【0007】

しかし、ICカードが暗号処理を行なっているときに、異常クロック、異常電圧、異常電磁波、異常温度等を用いて、故意にエラーを引き起こし、暗号で用いている鍵や秘密情報を取り出せる可能性があり、脅威となっている。John Wiley & Sons社 W.Rankl & W.Effing著「Smart Card Handbook」の8.5.1.2 Active protective mechanisms(263ページ)にこのような危険性が記載されている。この種の攻撃について、より詳細な議論は、Ross Anderson, Markus Gunter Kuhn:



“Tamper-Resistance --- a Cautionary Note”, the 2<sup>nd</sup> USENIX Workshop on Electric Commerce Proceedings, pp.1-11, 1996に書かれている。特にC R T (Chinese Remainder Theorem:中国人剰余定理)を用いたR S A暗号処理については、A.K.Lenstra氏のショートメモ “Memo on RSA Signature Generalization in the Presence of Faults”, 1999 に記載されている。このLenstra氏のショートメモについては、発明の実施形態で詳細を述べる。

## 【0008】

この種のアタックを防ぐ一つの方法は、特殊なハードウェアを内蔵することによって、異常な環境を検出することである。この理由から、現状流通しているI Cカードの多くは、各種の異常環境検出装置を内蔵している。

## 【0009】

ハードウェアを用いてこの種のアタックを防ぐ別の方法は、内部のレジスタなどにパリティビットをつける方法で、パリティチェックにより処理データに異常が検出されたときにリセットするなどして、異常な処理結果を返すことを防ぐものである。この方法は、主として大型の計算機のエラー対策として用いられているものであるが、一般にチップ面積の制約等の理由で、I Cカードには適用しないことが多い。

## 【0010】

しかしながら、異常環境検出装置を用いる対策は、検出装置の動的特性には限界があり、瞬間的な電源電圧の遮断や、クロックの変調等を検出することは容易でない。またパリティビットの検出では、2ビットが反転する等の誤作動を検出できない。

## 【0011】

## 【発明が解決しようとしている課題】

本発明の主たる課題は、異常環境検出装置やパリティ検出装置を用いることなく、I Cカード用チップにおいて生ずる誤作動を暗号処理方式に応じた方式で検出することである。本発明の着眼点は、暗号処理結果を出力する前に、再び復号化し、入力と同一であれば結果を出力し、異なれば結果を出力しないことにより、誤作動によって生じた処理結果を外部に出力しないことである。本発明の検出

方式は、異常環境検出装置やパリティ検出装置のように一般のプログラムを誤作動から保護することはできないが、もっとも重要な情報を操作する暗号処理部の誤動作を、異常環境検出装置やパリティ検出装置の検出限界を超えて検出することができる。

## 【0012】

本発明の目的は、上記の課題を解決することにある。

## 【0013】

## 【課題を解決するための手段】

ＩＣカードチップに代表される耐タンパー装置は、プログラムを格納するプログラム格納部、データを保存するデータ格納部を持つ記憶装置と、プログラムに従って所定の処理を実行しデータ処理を行う中央演算装置（ＣＰＵ）を持ち、プログラムは、ＣＰＵに実行の指示を与える処理命令から構成される一つ以上のデータ処理手段からなる情報処理装置として捉えることができる。ＩＣカードには、個人情報、電子マネーとしての機能など高いセキュリティを必要とする情報を格納する。そのため、ＩＣカードには暗号処理装置や、暗号ソフトウェアが内蔵されている。この意味で、ＩＣカードはデバイスとしては、暗号処理モジュールとして捉えることができる。暗号は大きく分けて二種類ある。一つは、暗号化に用いる鍵と復号化に用いる鍵が同一のもので、対称鍵暗号、または秘密鍵暗号と呼ばれるものである。もう一つは、暗号化に用いる鍵と復号化に用いる鍵が異なるもので、非対称鍵暗号、または公開鍵暗号と呼ばれるものである。後者は、特に電子署名等で用いられる方式である。

## 【0014】

本発明において、暗号処理中に生じた誤作動を検出する方法は、暗号処理結果を出力する前に、再び復号化し、入力と同一であれば結果を出力し、異なれば結果を出力しないことにより、誤作動によって生じた処理結果を外部に出力しないというものである。

## 【0015】

具体的には、対称鍵暗号、例えば、現在標準的に用いられているＤＥＳ暗号（例えば、岡本栄司著「暗号理論入門」共立出版，pp.33-50を参照）を用いる場合

、ICカードは、暗号文Cを受け取り、カードチップ内に格納されている秘密鍵Kを用いて、通常のDESの操作を行ない、平文 $Z = \text{INV\_DES}(C, K)$ を求める。DESは、16ラウンドからなる攪拌操作列であり、攪拌操作は、転字と換字から構成されている。このDESの攪拌操作を逆に溯る操作を行なうことにより、逆変換DESを構成することができる。従って、正しく復号処理 $\text{INV\_DES}(C, K)$ が行われれば、 $\text{DES}(Z, K) = C$ が成り立たなければならないはずである。そこで $\text{DES}(Z, K)$ の処理結果WをRAM等に格納した後、WとCを比較し、 $W = C$ となればZは正しい処理結果であることがわかるので、これを正しい処理結果として出力し、WがCと異なれば、出力しない。逆に平文を暗号化する場合は、復号化して確認することができることは言うまでもない。

## 【0016】

一方、非対称鍵暗号の場合、例えば、RSA暗号を例にとると、ICカードでは、（暗号化に用いられる公開鍵指数eは、通常、3又は、65537である）公開鍵指数e、公開モジュラスNを用いて、平文Mに対して、 $C = \text{RSA}(M, (e, N)) = M^e \bmod N$ を計算して、これを暗号文とする。この暗号文Cは、公開鍵情報 $J=(e, N)$ の持ち主によってICカードにて受信され、このICカードに保持されている秘密鍵指数xを用いて、 $\text{INV\_RSA}(C, x, J) = C^x \bmod N = M$ という操作で復号され、処理結果zを得る。一般にICカードのセキュリティにおいては、カードチップ内に格納されている秘密鍵指数xがアタックターゲットであり、この復号化処理において誤作動が生ずると、xに関する情報がカード外にリークする。これを守るために、計算結果Zをすぐに出力せず、RAM等に格納し、暗号化処理結果wとcを比較し、 $w = c$ であれば、Zは正しい処理結果であることがわかるので、これを正しい処理結果として出力し、wがcと異なれば、出力しない。

## 【0017】

以上を勘案すると、本発明の趣旨は、暗号化または復号化の操作に対し、その逆操作、すなわち暗号化に対しては復号化、復号化に対しては、暗号化の操作を行なって、元の結果が得られるかどうかを確認することにある。従って、暗号の種類が、DESであるか、RSAであるかと言った問題は本質的ではない。つまり、上記2種類の暗号以外に、他の秘密鍵暗号、公開鍵暗号に対しても同様の操

作一逆操作というプロセスで誤作動検出を行なうことができる。

【 0 0 1 8 】

【発明の実施形態】

本実施例では、秘密鍵暗号の代表例である D E S 暗号を例に取る。ここでは、秘密鍵暗号の代表例として D E S を例として採用するのみであって、D E S 以外の秘密鍵暗号においても同様に本発明を適用することができる。

【 0 0 1 9 】

図3は、D E S の基本構造を示す図である。D E S は、64ビットの平文を64ビットの鍵K（但し、このうち8ビットをパリティビットとして用いるので、実質の鍵長は、56ビットである）をビット置換302,304によって変形し、第一段目の部分鍵K1を生成し、置換302を行なった鍵ビットを左巡回シフト306,307で半々のビット毎に変形し、これをビット置換304と同じビット置換（PC-2）を施して部分鍵K2を生成する。これを繰り返す、最終的に、第16段目でも同様に左巡回シフト309,310で半々のビット毎に変形し、これをビット置換304と同じビット置換311を施して部分鍵K16を生成する。一方、平文は、初期置換IP301を施した後に、64ビットを32ビットずつに左右に分離される。この右半分を部分鍵K1と共にf関数303と呼ばれる非線型の変換に代入し、その結果と左半分のビットとビット毎の排他的論理和305を取って、第2ラウンドの右半分の32ビットとし、先の初期置換301の出力の右半分を第2ラウンドの左32ビットとして、以下同様の操作を繰り返す、最終的に第15ラウンド目の出力を部分鍵K16を用いて変形し、左右入れ替えた後、初期置換IPの逆置換313に代入して、その結果を64ビットの暗号文として出力する。

【 0 0 2 0 】

この復号変換INV\_DESは、図4のように構成することができる。図3との違いは、第16ラウンド目の処理から始めるということである。そのため先に左巡回シフト306,307,309,310で変形した部分を、逆に右巡回シフト406,407,409,410する。部分鍵はK16, K15, ..., K1というように暗号化変換とは逆に用いる。この操作は、ちょうど図3の処理を全て逆方向に行なうということに他ならない。

【 0 0 2 1 】

いま例えば、暗号化変換で、第16ラウンド目で、特定の処理ビットがエラーにより反転したとする。このとき、第16ラウンド目に使用されている部分鍵K16が何であるかによって、反転した際の処理結果が変化する。反転した処理結果とK16の関係を詳細に調べると、両者の間に数学的関係が現れる。これを複数の入力に対して連立して解くことにより、K16の候補を大幅に減らすことができる。K16が特定できれば、DESの鍵Kを決定するには、残りの8ビットを決定すればよいので、高々 $2^8 = 256$ 通りを試せば、正しい解を決定することができる。

## 【 0 0 2 2 】

DES暗号に対して誤作動を起こして解析する手法は、極めて複雑であるので、ここでは要点のみ示した。詳細は、国際会議CRYPTO'97にて発表された論文Eli Biham, Adi Shamir: "Differential Fault Analysis of Secret Key Cryptosystems", Springer-Verlag LNCS1294, pp513-525に書かれている。

## 【 0 0 2 3 】

このような攻撃を行なうには、アタッカーは、暗号化（または復号化）の結果を解析する必要がある。鍵K、平文Mに対する暗号化結果Zは通常RAMに一時的に格納され、ICカードのI/O端子を通して出力される。アタッカーは、暗号化処理中に異常電圧、異常クロック、異常電磁波などを印加し、エラーを引き起こす。従ってエラー注入が成功した場合のZは、一般に正しい処理結果DES(M, K)ではない別の値になっているはずである。逆に言えば、同じ値ではアタッカーは何の情報も得ることができない。

## 【 0 0 2 4 】

この性質を利用すれば、誤作動検出が可能となる。例えば、図5のような処理を行なえばよい。すなわち、I/Oポートから平文Mを受信し（ステップ501）、この平文MをRAMに格納する（ステップ502）。該平文MをICカード内のメモリ（通常はEEPROM）に格納されている秘密鍵Kと共にDES暗号化処理（ステップ503）を行なう。ステップ503の処理の結果ZをRAM上に格納し（ステップ504）、処理結果ZをDES復号化処理（ステップ505）を行ない、処理結果Wを得る。WとMを比較し（ステップ506）、両者が一致すればI/OポートからZを出力し（ステップ508）、WとMが一致しなければ、リセットする（ステップ507）。

D E S が K を固定された 64 ビットの数から 64 ビットの数への写像とみなせば、これは全単射であるから、Z が正しい DES(M, K) と一致する以外に、 $W = M$  となることはない。すなわち、D E S の処理結果に誤作動に起因するエラーがあれば、復号処理結果を見ることによって、必ずこのエラーを検出し、リセットがかかる。このときアタッカーは、アタックに必要となる誤った処理結果を得ることができず、アタックを実行することができない。これは、本発明の実施例の一つである。

## 【 0 0 2 5 】

復号処理の場合の誤作動検出も考え方は、全く同様である。すなわち、図 6 のように、I / O ポートから暗号文 C を受信し（ステップ 601）、この暗号文 C を R A M に格納する（ステップ 602）。暗号文 C を I C カード内のメモリ（通常は E E P R O M）に格納されている秘密鍵 K と共に D E S 復号化処理（ステップ 603）を行なう。ステップ 603 の処理の結果 Z を R A M 上に格納し（ステップ 604）、処理結果 Z を D E S 暗号化処理（ステップ 605）を行ない、処理結果 W を得る。W と C を比較し（ステップ 606）、両者が一致すれば I / O ポートから Z を出力し（ステップ 608）、W と C が一致しなければリセットする（ステップ 607）。すなわち、D E S の復号化処理結果に誤作動に起因するエラーがあれば、暗号化処理結果を見ることによって、必ずこのエラーを検出し、リセットがかかる。このときアタッカーは、アタックに必要となる誤った処理結果を得ることができず、アタックを実行することができない。これは、本発明の実施例の一つである。

## 【 0 0 2 6 】

このことを簡単に数値例で確認する。但し、D E S の計算は、手で追えるほど簡単ではなく、計算機を必要とするので、岡本栄司著「暗号理論入門」共立出版 p. 42 に掲載されている例を用いて説明する。簡単のため、復号化計算のチェックのみ実行してみる。

## 【 0 0 2 7 】

鍵 K = F234AEB545B1A830 (16 進数), 暗号文 C = 3CC0BAE8226AF5D1 (16 進数) に対する出力 M は、0952E3934CF0CB1E (16 進数) であることが知られている。この M が何らかの原因により、エラーを含み、1 ビット異なる値 0952E3934CF0CB1F (16 進数)

になってしまったと仮定する。これを再び暗号化してみると、9602F43C1283633Bとなる（この計算結果は表に書かれていない。実際に計算機を用いて計算する必要がある）。これは、明らかに本来の値C = 3CC0BAE8226AF5D1とは異なり、検出が成功する。

## 【 0 0 2 8 】

上記の一連の処理を見れば、秘密鍵暗号の種類がDESであることは、本発明において本質的ではなく、暗号化処理と、その復号化処理が与えられていれば、全く同様に行なうことができることがわかる。これを示したのが、図7及び図8である。すなわち、図7に示すように、I/Oポートから平文Mを受信し（ステップ701）、この平文MをRAMに格納する（ステップ702）。平文MをICカード内のメモリ（通常はEEPROM）に格納されている秘密鍵Kと共に暗号化処理（ステップ703）を行なう。ステップ703の処理の結果ZをRAM上に格納し（ステップ704）、処理結果Zを復号化処理（ステップ705）を行ない、処理結果Wを得る。WとMを比較し（ステップ706）、両者が一致すればI/OポートからZを出力し（ステップ708）、WとMが一致しなければ、リセットする（ステップ707）。すなわち、暗号化処理（ステップ703）の処理結果に誤作動に起因するエラーがあれば、復号処理結果を見ることによって、このエラーを検出し、リセットがかかる。このときアタッカーは、アタックに必要となる誤った処理結果を得ることができず、アタックを実行することができない。これは、本発明の実施例の一つである。

## 【 0 0 2 9 】

復号処理の場合の誤作動検出も考え方は、全く同様である。すなわち、図8のように、I/Oポートから暗号文Cを受信し（ステップ801）、この暗号文CをRAMに格納する（ステップ802）。暗号文CをICカード内のメモリ（通常はEEPROM）に格納されている秘密鍵Kと共にDES復号化処理（ステップ803）を行なう。ステップ803の処理の結果ZをRAM上に格納し（ステップ804）、処理結果ZをDES暗号化処理（ステップ805）を行ない、処理結果Wを得る。WとCを比較し（ステップ806）、両者が一致すればI/OポートからZを出力し（ステップ808）、WとMが一致しなければ、リセットする（ステップ807）。すなわち、D

ESの復号化処理結果に誤作動に起因するエラーがあれば、暗号化処理結果を見ることによって、このエラーを検出し、リセットがかかる。このときアタッカーは、アタックに必要となる誤った処理結果を得ることができず、アタックを実行することができない。これは、本発明の実施例の一つである。

## 【0030】

上記の実施例では、誤作動を検出した際にリセットを行なっているが、これは本発明の趣旨とは無関係であり、例えば、リセットは行なわず、暗号処理とは無関係な一定の値を出力したりしてもよいことは言うまでもない。

## 【0031】

また本発明の考え方は、暗号化処理、復号化処理の一部にも適用することができる場合がある。例えば、置換処理の最中にエラーが生じたかどうかを判定するために、この置換処理の逆置換処理を行なって、誤作動を検出することも可能である。

## 【0032】

次に、非対称鍵暗号の場合について説明する。非対称鍵暗号に対する誤作動を利用したアタックのうち、最も有名なのは、CRT（中国人剰余定理）を用いたRSA暗号処理に対するアタックである。この詳細はA.K.Lenstra氏のショートメモ“Memo on RSA Signature Generalization in the Presence of Faults”, 1999に記載されているが、ここでは、このアタックについて、その原理を説明し、理解の助けとする。RSA暗号および、CRTについては、岡本栄司著「暗号理論入門」（共立出版）や、A.J.Menezes, P.C. van Oorschot, S. A. Vanstone 著 Handbook of Applied Cryptography, (CRC-Press) などに詳しく記載されている。

## 【0033】

簡単にRSA暗号を説明する。RSA暗号では、大きな素数、例えば512ビットの2つの素数 $p, q$ の積 $N = pq$ と $N$ と互いに素な数 $e$ （ICカードでは、3や、65537が用いられることが多い）をとり、これを公開鍵として公開鍵簿に登録する。このとき、この公開鍵の持ち主Aに送信者Bは、1以上 $N-1$ 以下の数で表現されたデータ（平文） $M$ を、



$$y = M^e \bmod N$$

として暗号化して送信する。ここで、 $M^e$ は $M$ の $e$ 乗を表す記号とする。

この暗号文  $R$  を受け取った $A$ は、 $xe \bmod (p-1)(q-1) = 1$ となる秘密鍵  $x$  を用いて

$$y^x \bmod N$$

を計算する。ここで、 $(p-1)(q-1)$ は、 $N$ のオイラー関数の値 $f(N)$ である。これは、 $N$ と互いに素な自然数の個数に等しい。オイラーの定理によれば、

$$y^{((p-1)(q-1))} \bmod N = 1$$

が成り立つ。一方、 $xe = 1 + k(p-1)(q-1)$  ( $k$ は整数) と書くことができるので、

$$\begin{aligned} & y^x \bmod N \\ &= (M^e)^x \bmod N \\ &= M^{(ex)} \bmod N \\ &= M^{(1+k(p-1)(q-1))} \bmod N \\ &= M * M^{(k(p-1)(q-1))} \bmod N \\ &= M \end{aligned}$$

が成り立つ。従って、 $y^x \bmod N$ を計算することによって、持ち主 $A$ は、送信者 $B$ の平文  $M$  を復号することができる。この際、秘密鍵  $x$  を計算するのに、 $N$ の素因数  $p, q$  が用いられている。現在のところ、素因数分解を介さないで、 $x$ を計算する方法は知られておらず、大きな素数の積を因数分解することは、現実的でない時間が必要であるので、 $N$ を公開しても $A$ の秘密鍵は安全である。

#### 【0034】

ICカードでは、公開鍵指数 $e$ として、3や、65537が用いられることが多い。これは、暗号化の計算時間を短縮するという意味もあるが、 $e$ の値を知っても、直接的に秘密鍵指数 $x$ や $N$ の素因数が危険に曝されることはないという事情によるものである。

#### 【0035】

この計算法としては、アディション・チェイン方式などが採用される（上記「暗号理論入門」参照）ことが多いが、このようなアルゴリズムでは、処理が遅く

、ICカードを用いたトランザクションに要する時間がユーザの許容範囲を超えてしまう可能性がある。

【0036】

そこで、単純に $x$ 、 $N$ に対するべき乗剰余計算を行わずに、公開モジュラス $N$ の二つの素因数 $p, q$ に対するべき乗剰余計算結果から、 $M$ を導く方法が、CRTである。

【0037】

図9を用いて、CRTの処理を簡単に説明する。まず、計算に用いる値 $k=p^{-1} \bmod q$ 、 $x_p = x \bmod (p-1)$ 、 $x_q = x \bmod (q-1)$ の値を計算する。普通、これらの値はEEPROMに格納しておく。次にI/Oポートから暗号文 $y$ を受け取り（ステップ902）、この暗号文 $y$ を素因数 $p, q$ を法とする剰余 $y_p = y \bmod p$ 、 $y_q = y \bmod q$ を求め、これをRAMに格納する（ステップ903）。次に、二つのべき乗剰余計算：

$$C_p = y_p^{x_p} \bmod p, C_q = y_q^{x_q} \bmod q$$

を行なう（ステップ904、905）。次に、再結合計算：

$$S = (C_q - C_p) * k \bmod p$$

$$M = S * p + C_p$$

を行ない（ステップ906、907）、 $M$ を返す（ステップ908）。この $M$ が、実際の $y^x \bmod N$ に一致する。

【0038】

この事実を数値的に確認しておく。暗号文 $y = 79$ 、 $N = 187 (= 11 * 17)$ 、 $x = 107$ とする。この $x$ は、 $N$ のオイラー関数値 $(11-1)*(17-1)=160$ に関して、 $e = 3$ の逆数になっている。このとき、実際の値は、

$$\begin{aligned} M &= 79^{107} \bmod 187 \\ &= 79^{(5*3*7 + 2)} \bmod 187 \\ &= 79^2 * (79^5 \bmod 187)^{(3*7)} \bmod 187 \\ &= 79^2 * 10^{(3*7)} \bmod 187 \\ &= 79^2 * (10^3 \bmod 187)^7 \bmod 187 \\ &= 79^2 * (65^7 \bmod 187) \bmod 187 \end{aligned}$$

$$\begin{aligned}
 &= 79^2 * 142 \bmod 187 \\
 &= 29
 \end{aligned}$$

である。

【0039】

これをCRTを用いて計算する。 $11 * 14 \bmod 17 = 1$ であるから、 $k = 11^{(-1)} \bmod 17 = 14$ であり、 $x_p = 107 \bmod (11-1) = 7$ 、 $x_q = 107 \bmod (17-1) = 11$ である。また、 $y_p = 79 \bmod 11 = 2$ 、 $y_q = 79 \bmod 17 = 11$ となる。

$$C_p = 2^7 \bmod 11 = 7$$

$$C_q = 11^{11} \bmod 17 = 12$$

となるので、

$$S = (12 - 7) * 14 \bmod 17 = 2$$

$$M = 2 * 11 + 7 = 29$$

となり、先の値と一致する。

【0040】

CRTを用いると高速化できる理由は、べき乗剰余計算では、データ長さの3乗に比例して、計算量が増加するのに対して、CRTでは、データの長さが半分のを二つ計算するため、それぞれのべき乗剰余計算の計算量は、 $1/8$ で済み、これを二回実行しても、両者の計算量の合計は、 $1/8 * 2 = 1/4$ で済むからである。実際には、データの変換や、再結合計算を行なう必要があるので、4倍速にはならないが、速度は3倍程度に向上する。

【0041】

A.K.Lenstra氏が示したアタックの方法は以下の通りである。まず、ICカードを正常動作させて、正しい計算値Mを得る。次に計算中にエラーを注入し、図9における再結合計算部（ステップ907）において、Sが正しい値でなくなってしまうと仮定する。このエラーの入ったSの値を、S[ERROR]とし、対応する出力値をM[ERROR]すると、アタッカーは二つの値：

$$M = S * p + C_p$$

$$M[\text{ERROR}] = S[\text{ERROR}] * p + C_p$$

を得ることになる。両者の差は、



$$M[\text{ERROR}] - M = (S[\text{ERROR}] - S) * p$$

となる。つまり、結果の差は、素因数 $p$ の倍数である。

【0042】

従って、

$$p = \text{gcd}(M[\text{ERROR}] - M, N)$$

が成り立つ。ここで、 $\text{gcd}(A, B)$ は、 $A$ と $B$ の最大公約数である。

【0043】

エラーは、 $S$ の値を変化させ、 $C_p$ の値を変えないようなものなら何でもよい。つまり、 $y_q$ の計算値、 $C_q$ の計算値、 $(C_q - C_p) * k \bmod q$ の計算値のいずれかが本来の値と異なっていれば、上記アタックが成功する。

【0044】

実際、この方法で、モジュラスの因数分解ができることを数値例で示す。先に示した数値例を思い出そう。先の例では、暗号文 $y = 79$ ,  $N = 187 (= 11 * 17)$ ,  $x = 107$ とする。このとき、実際の値は、29であった。また $k = 11^{(-1)} \bmod 17 = 14$ ,  $x_p = 107 \bmod (11-1) = 7$ ,  $x_q = 107 \bmod (17-1) = 11$ ,  $y_p = 79 \bmod 11 = 2$ ,  $y_q = 79 \bmod 17 = 11$ であった。

【0045】

$C_q$ の計算が誤作動を起こし、11という値に変化してしまったと仮定する。 $C_p = 2^7 \bmod 11 = 7$ は正常値である。このとき、

$$S = (11 - 7) * 14 \bmod 17 = 5$$

となるので、

$$M[\text{ERROR}] = 5 * 11 + 7 = 62$$

が出力される。このとき、

$$\begin{aligned} & \text{gcd}(62 - 29, 187) \\ &= \text{gcd}(33, 187) = 11 \end{aligned}$$

となり、モジュラス $N$ の素因数11が得られる。

【0046】

本発明では、このような現象を次のように検出する。図10に示すように、まず、CRTの準備演算1001にて、 $k = p^{(-1)} \bmod q$ ,  $x_p = x \bmod (p-1)$ ,  $x_q = x \bmod (q-1)$

od (q-1)を計算し、メモリに格納しておく(ステップ1001)。次にI/Oポートから、暗号文yを受信し(ステップ1002)、この暗号文yをRAMに格納する(ステップ1003)。次に暗号文yに対して、CRTを用いたRSA復号計算  $y^d \bmod N$ を行なう(ステップ1004)。この演算結果ZをRAMに格納する(ステップ1005)。演算結果Zは、エラーを含んでいる可能性のあるものである。RAM上の演算結果Zに対して、暗号化計算  $Z^e \bmod N$ を行ない(ステップ1006)、暗号化結果Wと、RAM上にある暗号文yとが一致するかどうか比較し(ステップ1007)、一致すれば、I/Oポートに平文Zを出力し(ステップ1009)、一致しなければ、リセットする(ステップ1008)。これは、本発明の実施例の一つである。

【0047】

但し、これで、エラーが検出できるのは、yがモジュラスNと互いに素である場合である。このことは、オイラーの定理から容易にわかることである。もしもyがモジュラスNと互いに素でなければ、復号化結果を暗号化しても元に戻らない場合があり、この際、図10に示したエラー検出システムは、エラーがない場合でもリセットしてしまう。

【0048】

しかし、このようなことが起きる確率は、無視できる程度に小さい。実際、 $N=pq$ と互いに素な、N未満の自然数は、pの倍数がq-1個、qの倍数がp-1個であるから、 $p+q-2$ 個あるが、これは全体の

$$(p+q-2)/N = (p+q-2)/pq \approx (1/p) + (1/q)$$

に過ぎない。現在のRSA暗号の主流の鍵長は、1024ビットであるので、その素因数p,qは512ビットである。従って、上記確率は、ほぼ、 $2^{(-511)}$ であり、これは極めて小さな数であり、無視できる。

【0049】

本実施例においては、CRTを用いているが、エラーの検出は、CRTとは全く無関係であり、一般のRSAでもよい。さらに、より一般の公開鍵暗号でも利用可能である。以下、その一例として、楕円曲線上のRSA暗号を取り上げる。

【0050】

楕円曲線暗号については、考案者の一人が書いたN. コブリツ著(櫻井幸一訳

）「数論アルゴリズムと楕円暗号理論入門」（シュプリンガーフェアラーク東京）、楕円曲線上の演算については、I.H.シルバーマン・J.テイト著「楕円曲線論入門」（シュプリンガーフェアラーク東京）、また群、環、体等の代数系については、松坂和夫著「代数系入門」（岩波書店）に詳しい説明がある。

## 【0051】

説明に先立って、まず、楕円曲線暗号について、簡単に説明する。楕円曲線とは、体 $K$ の上で定義された3次多項式の零点集合であり、 $K$ の標数が2でない場合は、

$$y^2 = x^3 + ax^2 + bx + c$$

という標準形を持つ。標数が2の体の上では、

$$y^2 + cy = x^3 + ax + b \text{ または、}$$

$$y^2 + xy = x^3 + ax + b$$

という標準形を持つ曲線である。（いずれの場合も、後に説明する無限遠点 $O$ を含めて考える）。楕円曲線の形状は、図11のようなものになる。本発明において、標数が2であるか否かは、本質的ではないので、以下、簡単のため、標数が2でない場合について説明する。また暗号で必要なのは、有限体の場合のみであるので、その場合に限って説明する。有限個の元からなる体を有限体またはガロア体といい、その構造はよく知られている。その最も単純な構成法は以下の通りである。

## 【0052】

まず、素数 $p$ を法とする整数環の剰余環 $Z_p$ を考える。 $Z_p$ においては、0以外の元は逆を持つので、体の構造を持っている。これを素体といい、 $F_p$ と書く。これが最も原始的な有限体の例である。

## 【0053】

次に、 $F_p$ の元を係数に持つ多項式 $f(X)$ を考え、その零点のうち、 $F_p$ に含まれないものを $F_p$ に添加することによって、新しい体を構成することができる。これを、 $F_p$ の有限次代数拡大体という。 $F_p$ の有限次代数拡大体の元の個数は、 $p$ のべきになっていることが知られている。その元の個数を $q$ と書くとき、有限次代数拡大体を $F_q$ などと表示することがある。

## 【0054】

楕円曲線上の点の間には、演算を定めることができる。図12に示すように、楕円曲線上の二つの点、 $P, Q$ があるとき、この二点を通る直線を引き（ $P=Q$ のときは接線を引き）、この直線が再び楕円曲線と交わる点 $R$ を $x$ 軸に関して対称に折り返した点は、曲線の対称性から、再び楕円曲線上の点となる。この点を $P+Q$ と書き、 $P$ と $Q$ の「和」と定義する。交わる点がない場合は、架空の点として無限遠点というものを考え、この架空の点で交わっているものとみなす。無限遠点を $0$ と書く。また、楕円曲線上の点 $P$ と $x$ 軸に関して対称な位置にある点を $P$ の逆元といい、 $-P$ で表す。この「和」を用いて一点 $P$ を $k$ 個加えたものを、 $kP$ 、 $-P$ を $k$ 個加えたものを $-kP$ と書いて、 $P$ のスカラー倍という。これらの座標は、 $P, Q$ の座標の有理式で表すことができ、従って、一般の体の上でこの演算を考えることができる。この「加法」は、通常の加法と同様に、結合法則、交換法則が成立し、この加法に関して、無限遠点 $0$ は、通常の数での演算と同様にゼロの役割を果たし、 $-P$ は、 $P$ と加えると、 $0$ になる。これは楕円曲線上の加法演算が、可換群（アーベル群）の構造を持つことを示している。これをモデル・ヴェイユ群ということがある。楕円曲線 $E$ 、定義体 $F_q$ を固定したときのモデル・ヴェイユ群を、 $G(E/F_q)$ と書くことがある。 $G(E/F_q)$ の構造は非常に単純で、巡回群か、または二つの巡回群の直積と同型になることが知られている。

## 【0055】

一般に、 $kP=Q$ の値がわかっても、逆に $k$ の値を知るのは計算量が膨大になるため、容易でない。これを楕円曲線上の離散対数問題という。楕円曲線暗号では、楕円曲線上の離散対数問題が困難であることに基づいている。

## 【0056】

楕円曲線を利用した暗号方式には種々のものがあるが、ここでは、特に楕円 $RS$ A暗号方式を説明する。楕円 $RS$ A暗号においては、環上で楕円曲線を取り扱う必要がある。環上での楕円曲線においても、形式的に有限体上での場合と同じ式を用いてモデル・ヴェイユ群演算を行なうことができることが知られている。

## 【0057】

利用者は、二つの大きな素数 $p, q$  ( $p \equiv 2 \pmod{3}$ ,  $q \equiv 2 \pmod{3}$ ) を生成し、 $n = pq$ ,  $m = \text{lcm}(p+1, q+1)$  を求める。適当な  $e \in \mathbb{Z}_m (= \mathbb{Z}/m\mathbb{Z})$ ,  $\gcd(e, m) = 1$  を定め、 $d = e^{-1} \pmod{m}$  を計算する。 $(e, n)$  が公開され、 $d$  または、 $p, q$  を秘密鍵とする。

【0058】

暗号化は次のように行なう。 $M = (M_x, M_y) \in \mathbb{Z}_n \times \mathbb{Z}_n$  を平文とする。環  $\mathbb{Z}_n$  上の楕円曲線を

$$E: y^2 = x^3 + b$$

として、楕円曲線上の点の加算を考えると、点の加算式は、 $b$  の値に依存しないことがわかる。そこで、 $b = M_y^2 - M_x^3 \pmod{n}$  とおく。すると、 $M$  は、 $E$  上の点とみなすことができる。この設定上で、楕円曲線上の演算：

$$C = eM$$

を行なう。これが暗号化である。

【0059】

復号化は、

$$M = dC$$

とすればよい。この演算が復号になっていることは、RSA暗号の場合と同様に証明できるが、 $E$  の位数が  $p+1$  になっていることを利用する必要がある。詳しくは、例えば、岡本龍明・山本博資「現代暗号」産業図書を参照されたい。

【0060】

上記の楕円RSA暗号において、復号化操作におけるエラーを検出する方法について述べる。図13に示すように、まず、I/Oポートより、公開鍵 $e, n$ 及び暗号文 $C$ を受信し（ステップ1301）、この暗号文 $C$ をRAMに格納し（ステップ1302）、復号化計算（ステップ1303）にて、秘密鍵 $d$ を用いて $dC$ を計算する。 $dC$ には、エラーが含まれている可能性がある。この処理結果を $Z$ とし、この $Z$ に対して、暗号化計算（ステップ1305）にて、 $W = eZ$ を求める。もしも、 $Z$ が正しい結果であるならば、 $W$ は、 $C$ に等しくなければならない。そこで、 $W=C$ であれば、この $Z$ をI/Oポートに出力し（ステップ1308）、 $W \neq C$ でなければ、リセット（ステップ1307）を行なう。これは、本発明の実施例の一つである。

【0061】



以上に述べた処理方式は、抽象レベルでは、同一と考えられる処理を具現化したものであって、これらを、個々の暗号方式を超えて一般化することは自然なことである。

## 【 0 0 6 2 】

以下、図 1 4 を用いて上記エラー検出方法を抽象化したものを説明する。まず、I/Oポートより、公開鍵情報J及び暗号文Cを受信し（ステップ1401）、この暗号文CをRAMに格納し（ステップ1402）、復号化計算（ステップ1403）にて、秘密鍵情報Sを用いて、復号化結果 $D(C, S)$ を計算する。この復号化結果にはエラーが含まれている可能性がある。この処理結果をZとし、このZに対して、暗号化計算（ステップ1405）にて、 $W = E(Z, J)$ を求める。もしも、Zが正しい結果であるならば、Wは、Cに等しくなければならない。そこで、 $W=C$ であれば、このZをI/Oポートに出力し（ステップ1408）、 $W=C$ でなければ、リセット（ステップ1407）を行なう。

## 【 0 0 6 3 】

図 1 4 に示した処理は、どのような非対称暗号に対しても適用できるわけではないことに注意しておく。実際、楕円ElGamal暗号などでは、単純に逆算できないため、本発明の方式は適用できない。

## 【 0 0 6 4 】

## 【発明の効果】

以上述べたように本発明によれば、暗号化または復号化の操作に対し、その逆操作、すなわち暗号化に対しては復号化、復号化に対しては暗号化の操作を行なって、元の結果が得られるかどうか確認するので、故障検出によるICカードなどへの攻撃に対抗することができる。

## 【図面の簡単な説明】

## 【図 1】

ICカードの概観及び端子を示す図である。

## 【図 2】

マイクロコンピュータの構成を示す図である。

## 【図 3】

DES暗号化処理方式を説明する図である。

【図 4】

DES復号化処理方式を説明する図である。

【図 5】

DES暗号化に対するエラー検出方法の実施例の処理手順を示す図である。

【図 6】

DES復号化に対するエラー検出方法の実施例の処理手順を示す図である。

【図 7】

一般の秘密鍵暗号の暗号化に対するエラー検出方法の実施例の処理手順を示す図である。

【図 8】

一般の秘密鍵暗号の復号化に対するエラー検出方法の実施例の処理手順を示す図である。

【図 9】

CRT（中国人剰余定理）を用いたRSAべき乗剰余計算の処理手順を示す図である。

【図 1 0】

CRT（中国人剰余定理）を用いたRSA復号化計算に対するエラー検出方法の実施例の処理手順を示す図である。

【図 1 1】

楕円曲線の形状を示す図である。

【図 1 2】

楕円曲線上の加法を説明する図である。

【図 1 3】

楕円RSA暗号における復号化操作に対するエラー検出方法の実施例の処理手順を示す図である。

【図 1 4】

より一般の非対称鍵暗号における復号化操作に対するエラー検出方法の実施例の処理手順を示す図である。

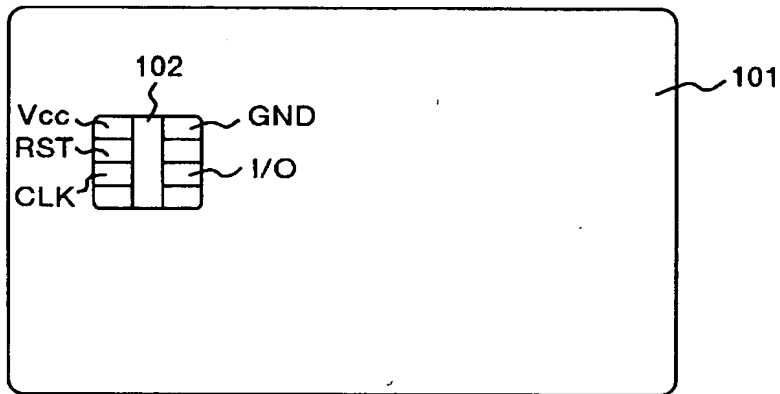
【符号の説明】

601～608：ステップ（復号化処理後の暗号化による確認）、1301～1308：（復号化処理後の暗号化による確認）

【書類名】 図面

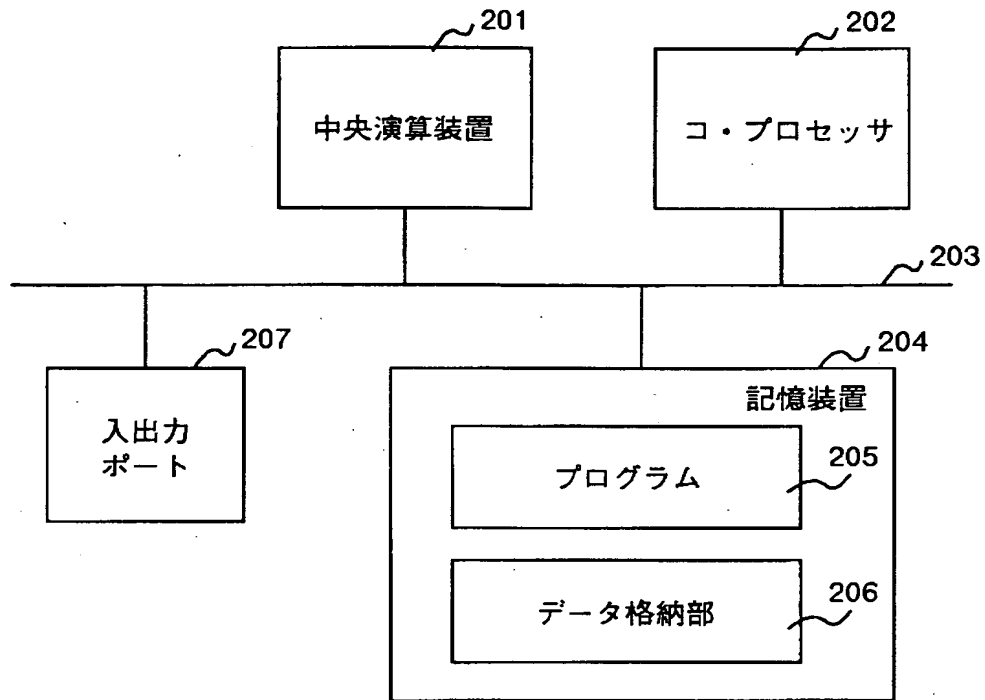
【図 1】

図 1



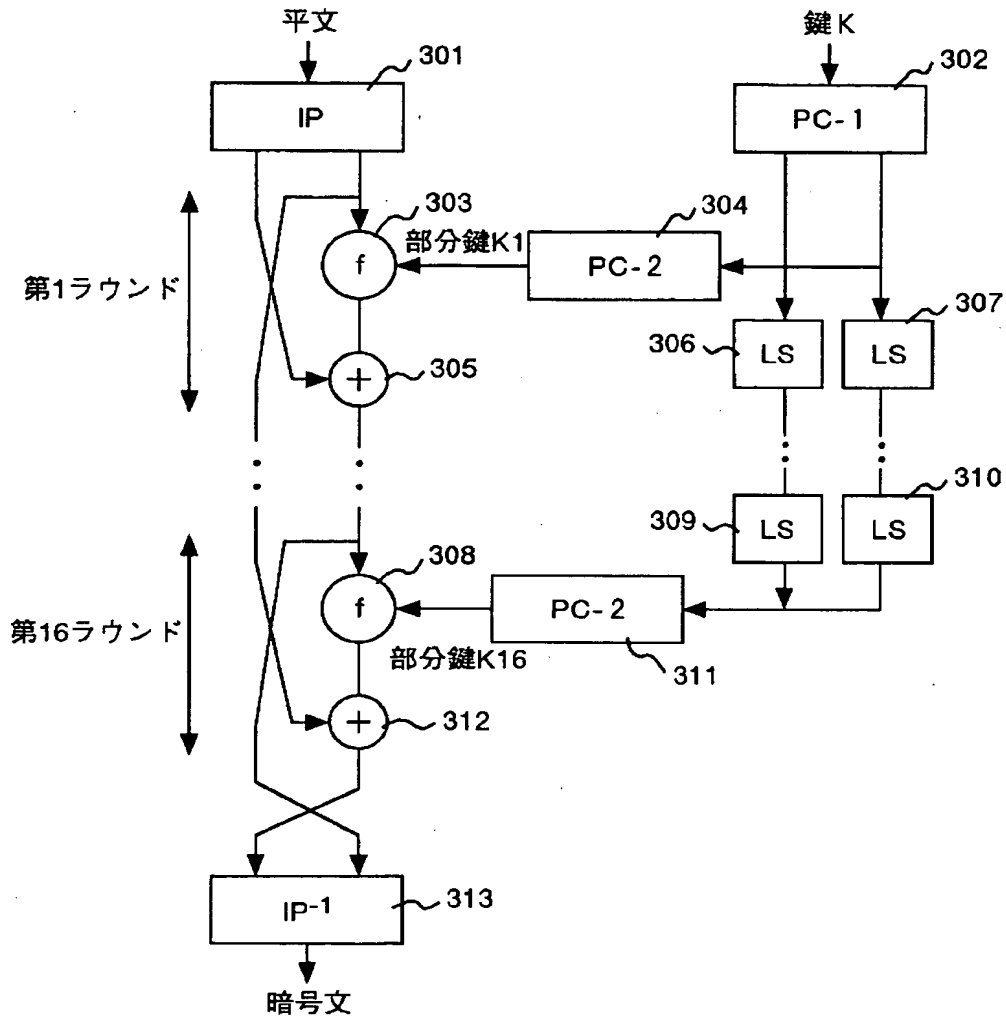
【図 2】

図 2



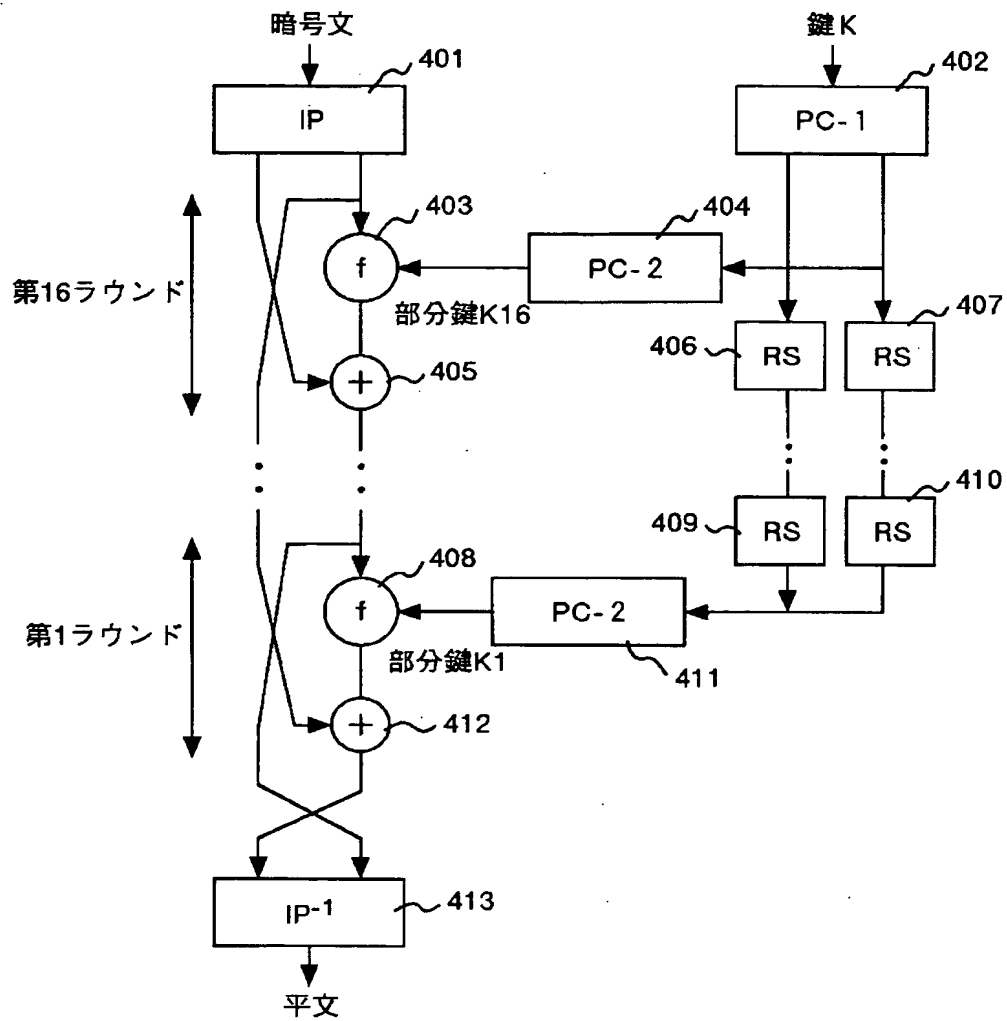
【図 3】

図 3



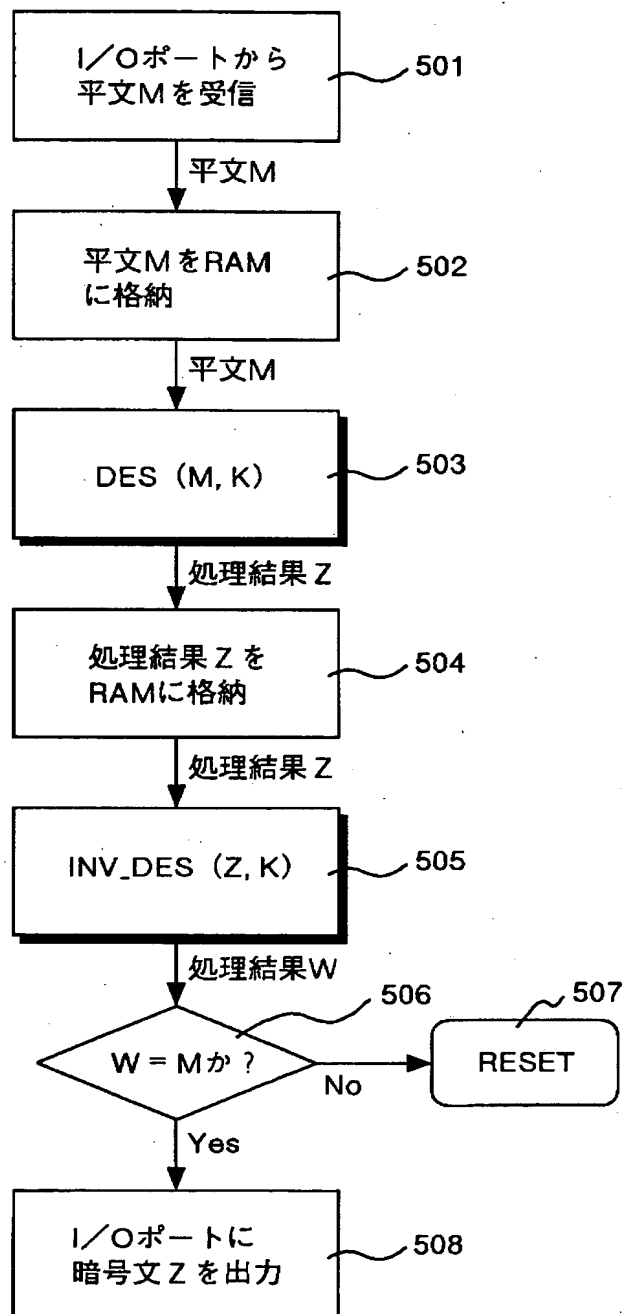
【図4】

図 4



【図 5】

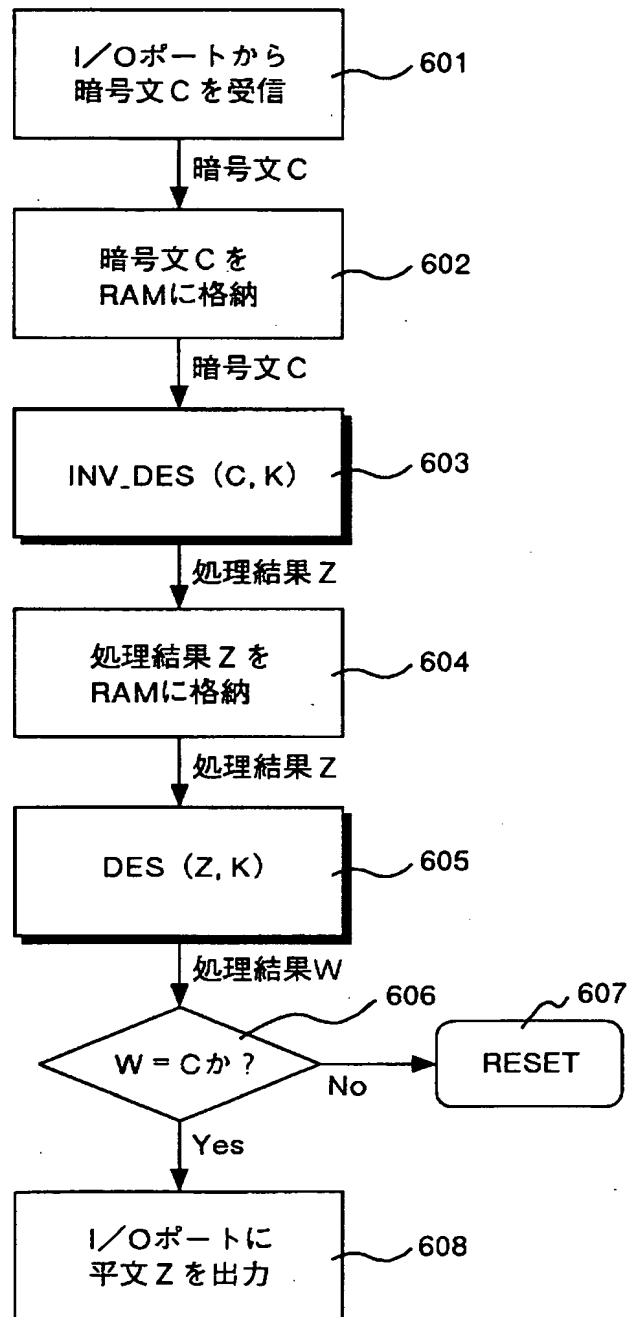
図 5





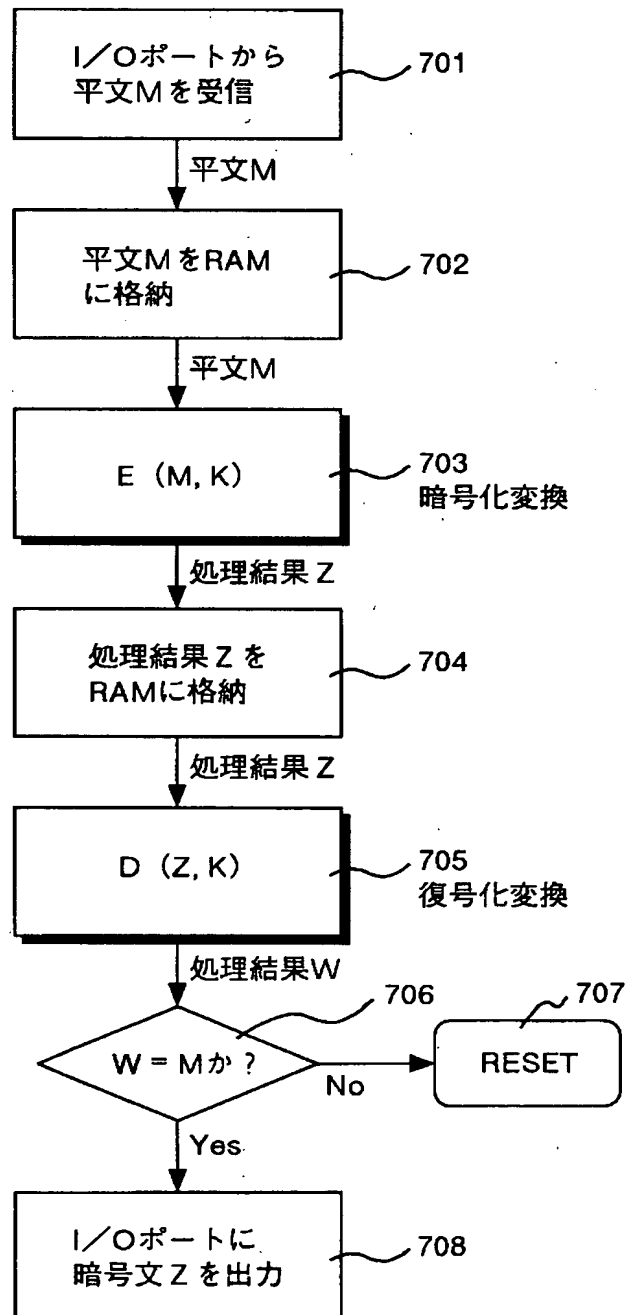
【図 6】

図 6



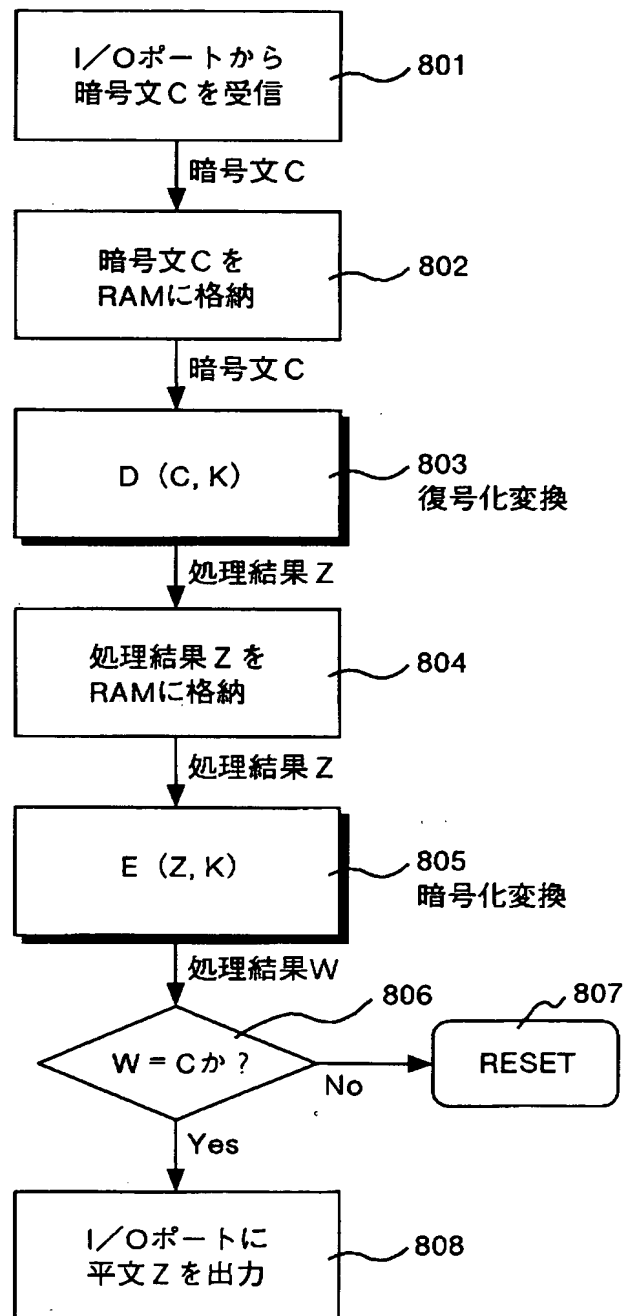
【図 7】

図 7



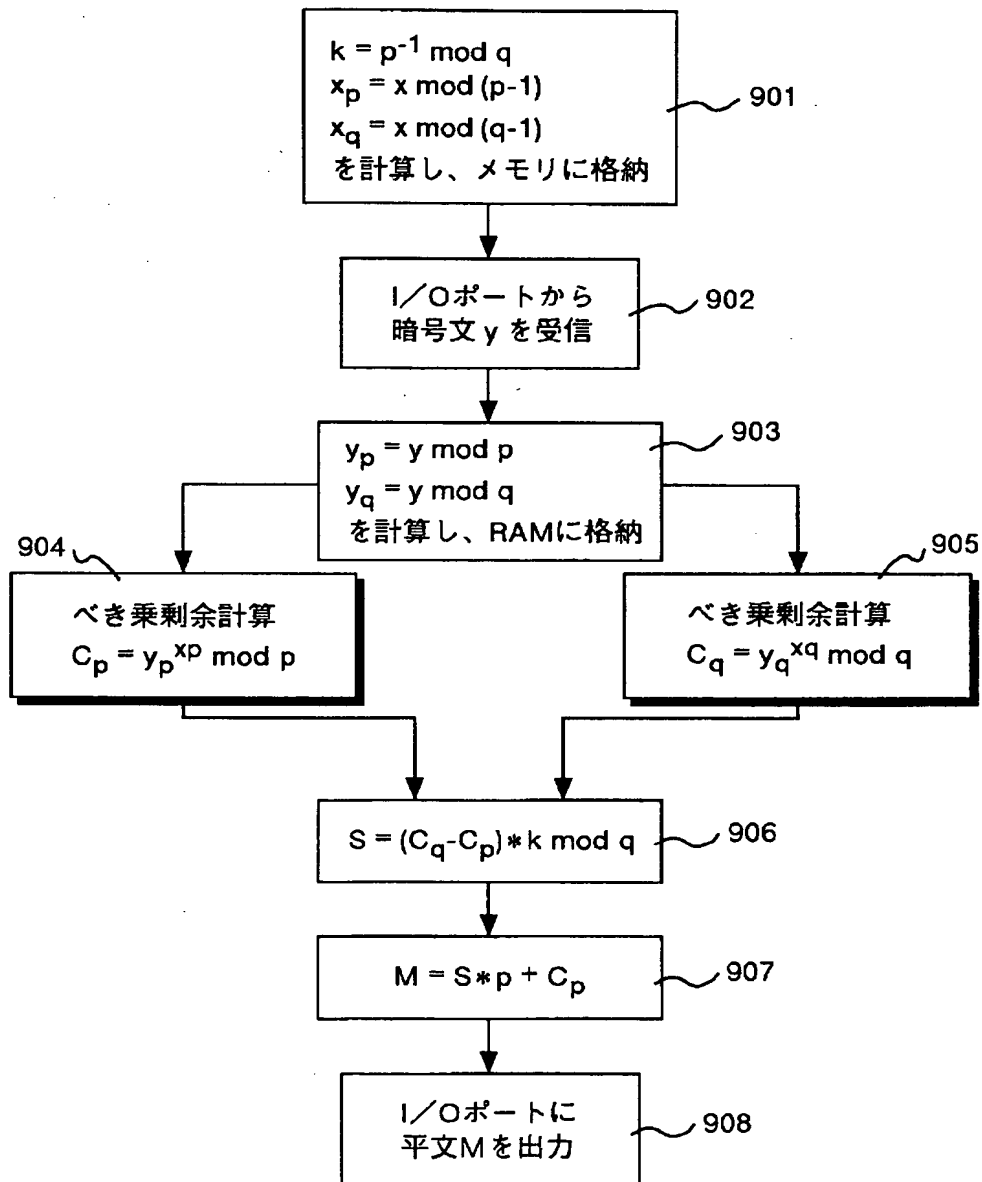
【図 8】

図 8



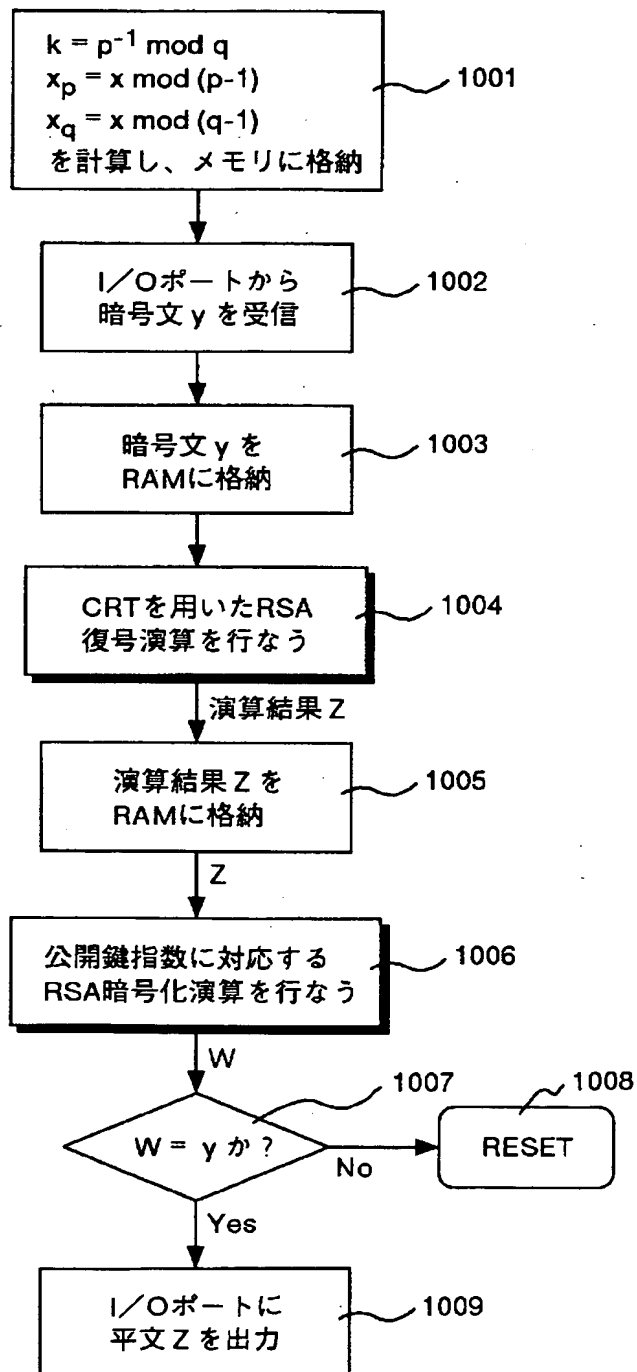
【図 9】

図 9



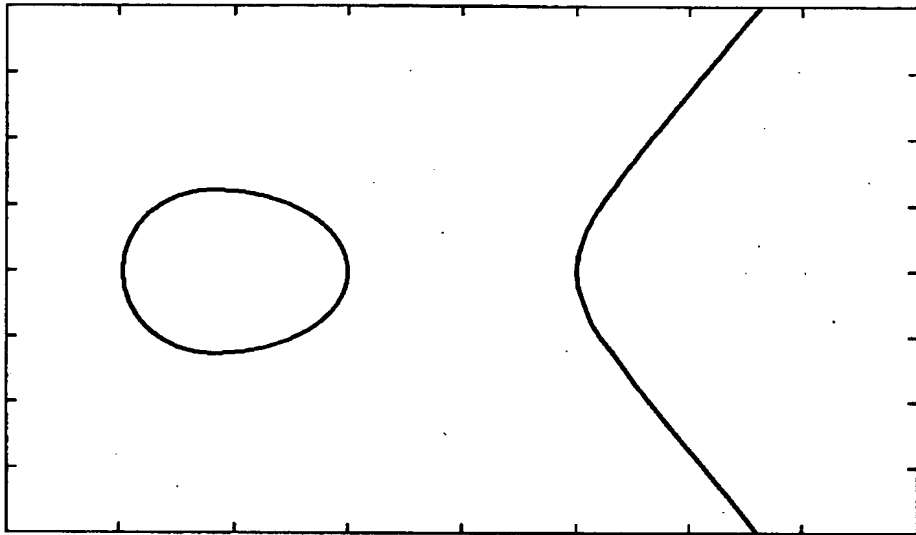
【図10】

図 10



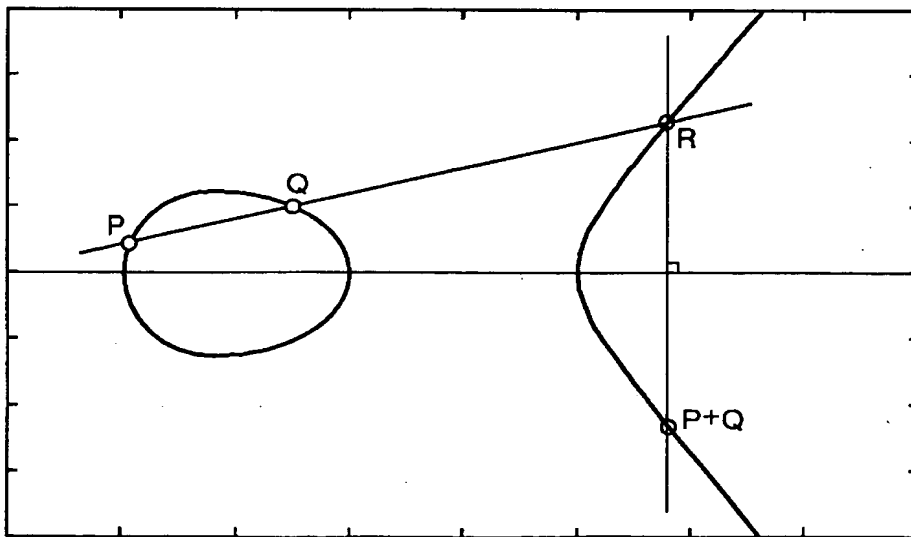
【図11】

図 11



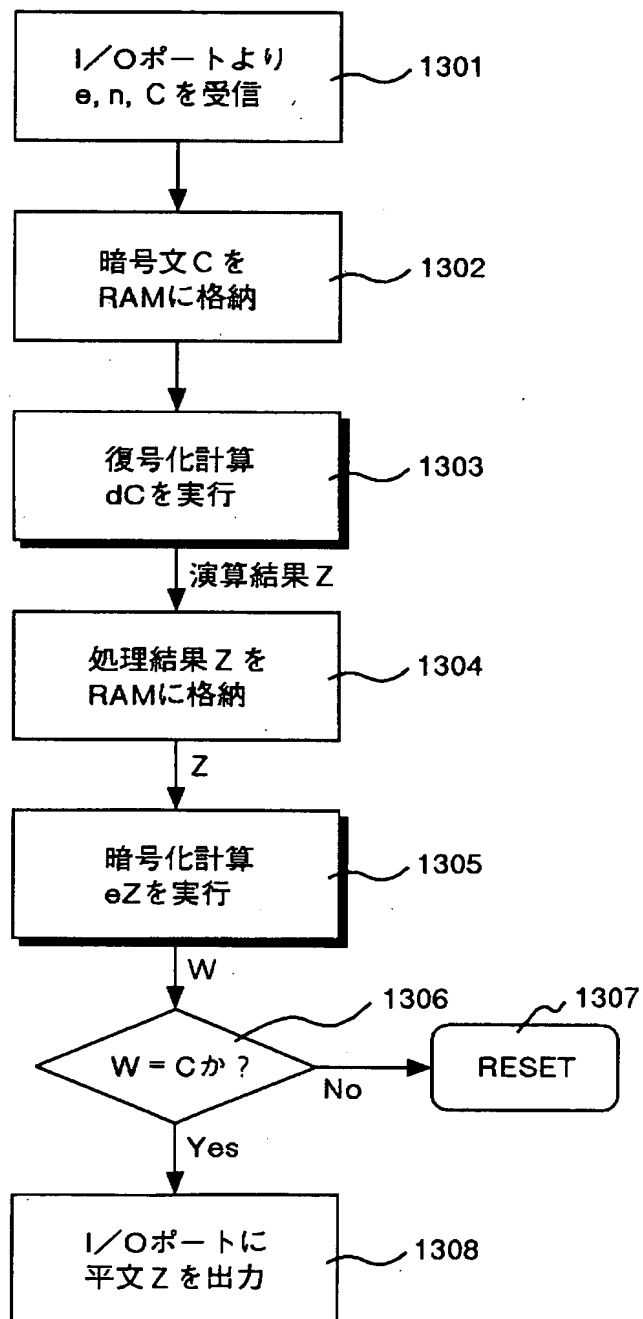
【図12】

図 12



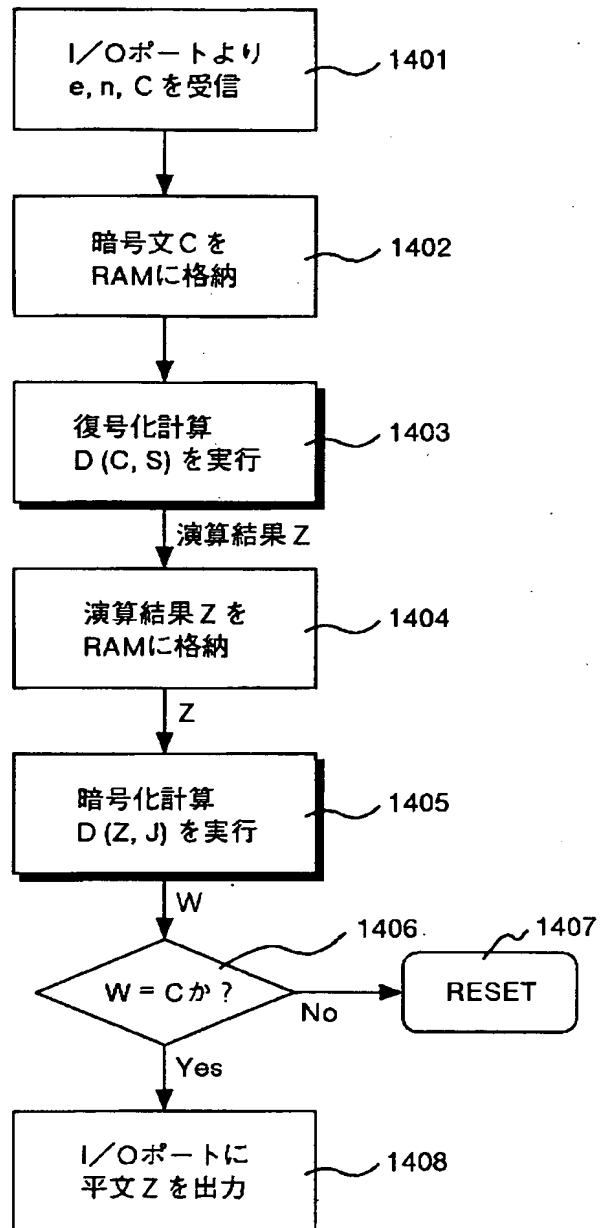
【図 13】

図 13



【図 14】

図 14





【書類名】 要約書

【要約】

【課題】 ICカードなど内部で暗号処理を行なう装置に対して故意にエラーを引き起こし、秘密情報を取り出すような攻撃に対抗する暗号処理方法を提供する。

【解決手段】 ICカードなどのI/Oポートから暗号文Cを受信し（ステップ601）、この暗号文CをRAMに格納し（ステップ602）、暗号文Cに対して復号化処理を行ない（ステップ603）、その処理結果ZをRAMに格納する（ステップ604）。処理結果Zに対して暗号化処理を行ない（ステップ605）、その処理結果Wと元の暗号文Cとを比較して（ステップ606）、一致すればI/Oポートに平文Zを出力する（ステップ608）。両者が不一致であればリセットする（ステップ607）。

【選択図】 図6

特2001-058087

出 願 人 履 歴 情 報

識別番号 [000005108]

1. 変更年月日	1990年 8月31日
[変更理由]	新規登録
住 所	東京都千代田区神田駿河台4丁目6番地
氏 名	株式会社日立製作所